

f is $O(g) \equiv$

$$\exists c, k > 0 \quad \text{s.t.} \quad \forall n \geq k \quad f(n) \leq c \cdot g(n)$$

$n \equiv$ length of input

$f \equiv$ describes runtime

$$\underbrace{2n^2 + 3n + 5}_f \quad \text{is} \quad O(\underbrace{n^2}_g)$$

w.t.s $\exists c, k > 0$ s.t.

$$\forall n \geq k \quad 2n^2 + 3n + 5 \leq c \cdot n^2$$

$$k=1 \quad c=30$$

$$\text{w.t.s} \quad 2n^2 + 3n + 5 \leq 30n^2 \quad \forall n \geq 1$$

$$\underline{2n^2} + 3n + 5 \leq \underline{10n^2} + 10n^2 + 10n^2$$

$$a + b + c$$

$$d + e + f$$

$$a \leq d$$

$$b \leq e$$

$$c \leq f$$

$$\Rightarrow a + b + c \leq d + e + f$$

✓ is $2n^2 \leq 10n^2 \quad \forall n \geq 1$?

$$1 \leq 5$$

✓ $3n \leq 10n^2 \quad \forall n \geq 1$

$$3 \leq 10n$$

✓ $5 \leq 10n^2$

$$2n^2 + 3n + 5 \leq 30n^2 \quad \blacksquare$$

2nd Proof w.t.s.

$$2n^2 + 3n + 5 \leq C \cdot n^2 \quad \text{for some } C \text{ \& all } n \geq 1$$

$$2n^2 \leq 2n^2$$

$$3n \leq 3n^2$$

$$5 \leq 5n^2$$

$$\forall n \geq 1$$

$$2n^2 + 3n + 5 \leq 10n^2 \quad \forall n \geq 1$$

$$C=10 \quad \blacksquare$$

Function bubble_sort (array, len) :

for j = 0 ... n-1:

for i = 0 ... n-j-1:

if array[i] > array[i+1]:

temp = array[i]

array[i] = array[i+1]

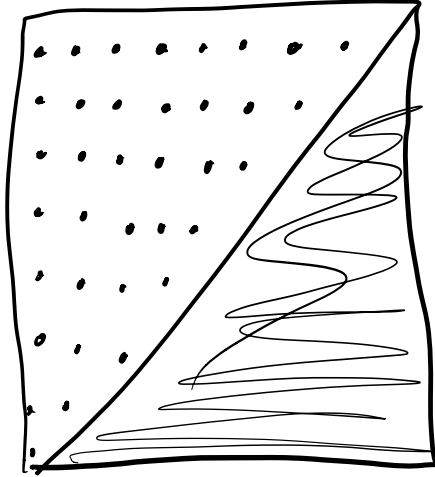
array[i+1] = temp



const.

$$c \cdot n \cdot n = O(n^2)$$

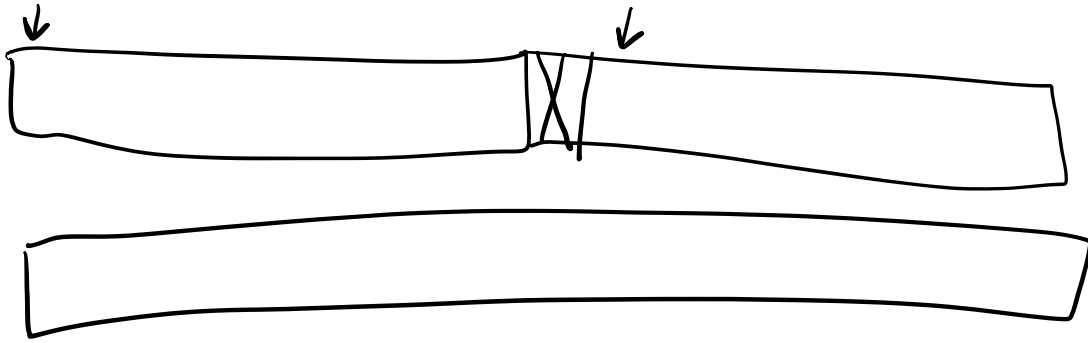
$$n + (n-1) + (n-2) \dots + 1 \leftarrow \frac{n(n+1)}{2}$$



$$\frac{n^2}{2} + \frac{n}{2}$$

↓

$$O(n^2)$$



Function `merge_sort(arr, len)`

if `len == 1`:
return `arr`

`arr1` = `⌈first half of arr⌋`

`arr2` = `⌊second half of arr⌋`

`sorted1` = `merge_sort(arr1, ⌈len/2⌉)`

`sorted2` = `merge_sort(arr2, ⌊len/2⌋)`

`sorted` = `merge(sorted1, ⌈len/2, sorted2, ⌊len/2⌋)`

return `sorted`

```

Function merge (arr 1 , len1 , arr 2 , len 2 ) :
    out_arr = array of length (len1 + len2)
    pos 1 = 0
    pos 2 = 0
    while ( pos 1 < len 1 and pos 2 < len 2 ) :
        if ( arr 1 [ pos 1 ] ≤ arr 2 [ pos 2 ] ) :
            out_arr [ pos 1 + pos 2 ] = arr 1 [ pos 1 ]
            pos 1 += 1
        else :
            out_arr [ pos 1 + pos 2 ] = arr 2 [ pos 2 ]
            pos 2 += 1
        end if
    end while
    while ( pos 1 < len 1 ) :
        out_arr [ pos 1 + pos 2 ] = arr 1 [ pos 1 ]
        pos 1 ++
    end while
    while ( pos 2 < len 2 ) :
        out_arr [ pos 1 + pos 2 ] = arr 2 [ pos 2 ]
        pos 2 ++
    end while

```

```
end while  
return out_arr
```